Accessible Games for Blind Children, Empowered by Binaural Sound

Konstantinos Drossos
Lab of AV Signal Processing
Dpt. of Audiovisual Arts
Ionian University
Corfu, Greece
kdrosos@ionio.gr

Nikolaos Zormpas
Science For You, N.P.C. - SciFy
17, Amphiktyonos str., Thissio
Athens, Greece
nickzorb@scify.org

George Giannakopoulos
Science For You, N.P.C. - SciFy
17, Amphiktyonos str., Thissio
Athens, Greece
ggianna@scify.org

Andreas Floros
Lab of AV Signal Processing
Dpt. of Audiovisual Arts
Ionian University
Corfu, Greece
floros@ionio.gr

ABSTRACT

Accessible games have been researched and developed for many years, however, blind people still have very limited access and knowledge of them. This can pose a serious limitation, especially for blind children, since in recent years electronic games have become one of the most common and wide spread means of entertainment and socialization. For our implementation we use binaural technology which allows the player to hear and navigate the game space by adding localization information to the game sounds. With our implementation and user studies we provide insight on what constitutes an accessible game for blind people as well as a functional game engine for such games. The game engine developed allows the quick development of games for the visually impaired. Our work provides a good starting point for future developments on the field and, as the user studies show, was very well perceived by the visually impaired children that tried it.

Categories and Subject Descriptors

H.5.2 [**User interfaces**]: Auditory feedback

General Terms

Human factors

Keywords

Audio only games, auditory interface, binaural processing, games for the visually impaired

1. INTRODUCTION

PETRA '15, July 01 - 03 2015, Island of Corfu, Greece.
ACM 978-1-4503-3452-5/15/07...$15.00.
DOI: http://dx.doi.org/10.1145/2769493.2769546

The advances of technology have resulted in the wide spread of video games [8], which often are not fully accessible. Most games can be played on a range of devices, starting from a common personal computer to a mobile phone. On any platform or device, a video game is, usually, heavily based on the visual communication channel. Many of the game aspects, e.g. player-game interaction, scenery and scenario, guidelines, tutorials and others, are primarily communicated through colors, shapes, text and visual objects. The audio communication channel appears to be under-utilized, even though it has been shown to be effective as an interface [4], as a means to entertain [14], and as a means to provide guidance [16].

Children, adolescents, young and mature adults play games for entertainment. People with vision impairments may, however, find it impossible to play games which rely on visual feedback. In such cases, video games can be rendered obsolete. Especially in the case of visually-impaired children, the impact of not being able to play games with one's peers is likely to have significant social side-effects, e.g. alienation. Such problems can be rectified through the use of novel interface approaches.

Audio games [8] are games, played in an electronic device, e.g. personal computer or mobile device, that employ the audio channel as their main interface. Such games utilize auditory displays, instead of visual elements, for correctly and intentionally conveying all the aforementioned types of information. The latter fact depicts with clarity that the key element of audio games is the sonic design, in terms of accurate and efficient human–computer interaction. As mentioned in [8], audio games can be classified in two categories:

Legacy Game Remakes which may be audio-only versions of legacy games, including typical grid–based games [8], e.g. Tic-Tac-Toe, which are also useful for visually impaired people [15];

Novel, Audio-Only Games that are based on new ideas and developed from scratch. In the latter category are situated audio games like Papa Sangra II or BlindSide [17, 6].

In the current work we extend a previously developed, audio-only, prototype version of the legacy game Tic-Tac-Toe [8] by also including sonic displays in order to render it attractive for blind children. We performed user tests with blind children in order to evaluate the newly added sounds and the audio interface in general. The rest of the paper is as follows: in Section 2 we provide a brief overview of the previous work in audio games targeted to visually impaired people; in Section 3 we present the developed audio game and in Section 4 we present the application of the use case; Section 5 concludes the current paper.

2. RELATED WORK

As can be inferred, audio interaction is a key component in the current work. The perceived sound by the user must communicate all the necessary elements of the game in order to replace the visual channel without using verbal sounds. The latter is termed sonification [13]. To that aim, in the existing literature two main concepts have emerged:

Auditory / Sonic Displays Auditory displays employ recognizable sounds, emerging from various every-day life situations, e.g. ringings, steps etc. With these, the transfer of information is performed by correct assignment of the communicated pieces of information to the actual semantic content of the auditory icon [8, 11]. In other words, the semantic content of an auditory icon such as the sound of an alarm clock could be mapped to a reminder about an event one needs to attend soon.

Earcons Earcons are artificial sounds synthesized from multiple single-audio motifs. These motifs can be altered, in terms of amplitude, frequency, duration and timbre [1]. Composite earcons can also be created by combining multiple sound motives or earcons in a single sound file. These composite earcons can be used to communicate concurrent information when needed. The concurrent presentation of information can be further enhanced by the utilization of audio spatial positioning, e.g. binaural rendering which is elaborated below.

Binaural technology allows the augmentation of spatial positioning of sound with the usage of a simple pair of headphones. The audio stimulus can be virtually placed around the listener's head, covering a range of 360^o in the horizontal axis. Additionally, there is also the capability for elevation typically ranging from $\approx 90^o$ to $\approx -90^o$, with 0^o being the receiver's head position in the vertical axes. Besides the construction of audio virtual environment, binaural technology has also been used recently in many aspects that employ sound localization, e.g. hearing aids [2], dereverberation [12], stereo recording enhancements [5] and emotion recognition from sound events [3].

One of the earliest works on audio games [18] presents what is probably the first use case study of an audio game. It is shown that audio interfaces can be used to express complex auditory spaces by using basic stereo sound localization techniques and sound manipulations. Through the years several techniques have been used to better emulate the auditory spaces [7], e.g. pitch shifting or volume manipulation. Studies have also tested the use of haptic feedback devices [19] with significant success. Using a second feedback channel, besides audio, allows the creation of more complicated games by allowing concurrent transfer of more information.

Designing an audio game, however, is not a trivial task. It is important to plan the entire game concept around an auditory experience [9]: this makes the game immersive and usable. Our implementation, as discussed above, is based on binaural audio, which has had limited use in the past due to its implementation challenges. The game was designed specifically to meet the requirements of visually impaired children using this sound technology.

In this work, we implement both auditory icons and earcons in order to communicate the aspects of the Tic-Tac-Toe game to blind children. To that end, binaural rendering is also employed for the spatial positioning of the used audio stimuli, as well as to transfer the physical aspects of the Tic-Tac-Toe board — i.e. the position of the 9 squares that the board has — through the audio channel. Our approach goes beyond the implementation of a single game. The game engine developed and described in this paper supports any of the audio techniques discussed and provides a starting point for the creation of complex and accessible audio games.

3. CHALLENGES AND IMPLEMENTATION

Before delving in the details of our implementation, we briefly describe the basics of a Tic-Tac-Toe game. Then we overview the basic challanges that the game design had to tackle, based on the requirement analysis that was conducted early in the implementation process (more information on the analysis is provided in Section 4).

The Tic-Tac-Toe game is a strategy game, positioned in a 3 by 3 grid. Each player chooses a symbol, usually from the set $\{X, O\}$. The game is played in turns. On each turn a player positions his symbol to an available (i.e. not previously assigned) square of the grid. Each player aims to fill in a line, row or diagonal of the grid with his own symbol only. The player who first achieves such a state in the game wins. If no player can achieve such a state and the grid squares have been fully assigned, then the game ends at a draw.

Even though the game appears to be easy to understand, it heavily relies on the visual channel to communicate information. The main challenges faced during game design were as follows:

Spatial understanding The 3 by 3 grid can be understood and communicated instantly to someone who has a view of the grid. People who have limited experience of shapes and grids and limited vision, however, find understanding the relative position of grid cells more challenging. This appeared to specially hold for children, that may even have spatial understanding limitations due to their visual disability.

Game state Understanding the game state per turn can be easy for a seeing person, but a visually impaired player needs to constantly hold a cognitive representation of the assigned squares to understand threats and apply a strategy. This requires an easy way to navigate and query the state of the game board in order to support and update the congitive representation.

Positioning Similarly to the previous challenge, players need to be aware of available squares to be able to position their symbol during their turn. Again, the communication of this subset of the game state (available squares) is more challenging when the visual channel cannot be employed.

Rules understanding Beyond the two previous challenges, understanding what constitutes an end state of the game (either a win, a loss or a draw) implies understanding of shapes (horizontal line, vertical line, diagonal). The same stands for a threat, which essentially is the state where an opponent has almost formed a 3-symbol line (vertically, horizontally or diagonally) and there exists at least one unassigned position they can use to complete the line.

Partial vision We needed to be able to support persons with partial vision (not totally impaired), by allowing them to combine this vision with the auditory channel to play the game.

The design we used to tackle the above challenges in the Tic-Tac-Toe setting can be focused on three separate components, a Game Engine, an Audio Engine, and the implementation of the Tic-Tac-Toe itself. The game engine controls the flow of information between the User Interface and the game world, represents the game world, holds the current Game State and facilitates the definition and use of Game Rules. The audio engine handles sound settings, serves as the audio Application Program Interface and implements the sonic design according to the provided sonic interfaces. We elaborate on these components and concepts in the following paragraphs.

3.1 Game Engine

The game engine developed in LEAP Tic-Tac-Toe accommodates the need for quick, accessible game production. It builds upon the notion of a Game State to describe the game world over time and uses several components to facilitate the interaction between the user and the game world as can be seen in Figure 1. These components include an abstraction differentiating between our implementation of the game and its rendering, allowing the integration with other game engines such as Unity[1].

The Game State is a representation of the game world in a specific point in time. It stores all the needed information, e.g. currently active game objects, actors, and game statistics. Game objects represent everything that exists inside the playing field, either visible, e.g. walls and players, or invisible, e.g. sounds. The most common type of game objects are the static objects. Static objects store their location inside the game world and are also accompanied by a description of their visual representation. These static objects define the current condition of the game stage and do not contain any logic. On the other hand, actors, which belong to a special subcategory of game objects, employ logic. Each actor represents an actively moving and interacting object, e.g. a player's pointer or a bouncing ball. To achieve that, each actor contains its own logic that controls reactions to user input, e.g. movement commands, and how the interacts with other game objects, e.g. other actors and static objects.

[1] See `http://unity3d.com/` for more information.

Sound objects are also considered static objects as they "exist" in a specific point inside the game world. Sound objects, however, have a limited life span and are automatically removed when the sound they represent finishes its playback. Sound objects were given special care to accommodate the auditory needs of a fully accessible game for the visually impaired. Sound objects were divided into three subcategories with each game state maintaining a different list for each one. These subcategories are standard sound objects, interruptible sound objects, and blocking sound objects.

Standard Sound Objects represent sounds that play normally until finished without interacting with the rest of the game, e.g. background music.

Interruptible Sound Objects are used to represent sounds of usually lesser importance that carry redundant information or cosmetic value. These are automatically discarded whenever a user action triggers the generation of a new interruptible or blocking sound object. This allows the existence of cosmetic sounds or redundant information messages that do not block any essential flow of information.

Blocking Sound Objects are sounds that carry important information and require the user's full attention. Whenever the generation of a blocking sound object is triggered any interruptible sound objects are automatically discarded. Additionally, while a blocking sound object is being played, any user action is prevented if it could lead to the generation of a new blocking or interruptible sound object.

These three sound object types, when used together, allow the creation of the audio interface for any game component (stage). These game components can be as simple as a basic menu or as complicated as an advanced tutorial with intricate game logic. The actual sounds that correspond to each sound object are acquired through the audio engine's audio Application Programming Interface (API) as will be discussed bellow.

To control the flow of information, between the users and the game state, the game engine employees the following components:

Game Manager Each instance of the Game Manager contains a game state and a set of rules. Most games require the implementation of several different stages with similar rule-sets and initial states. This implementation aims to represent each game stage as an

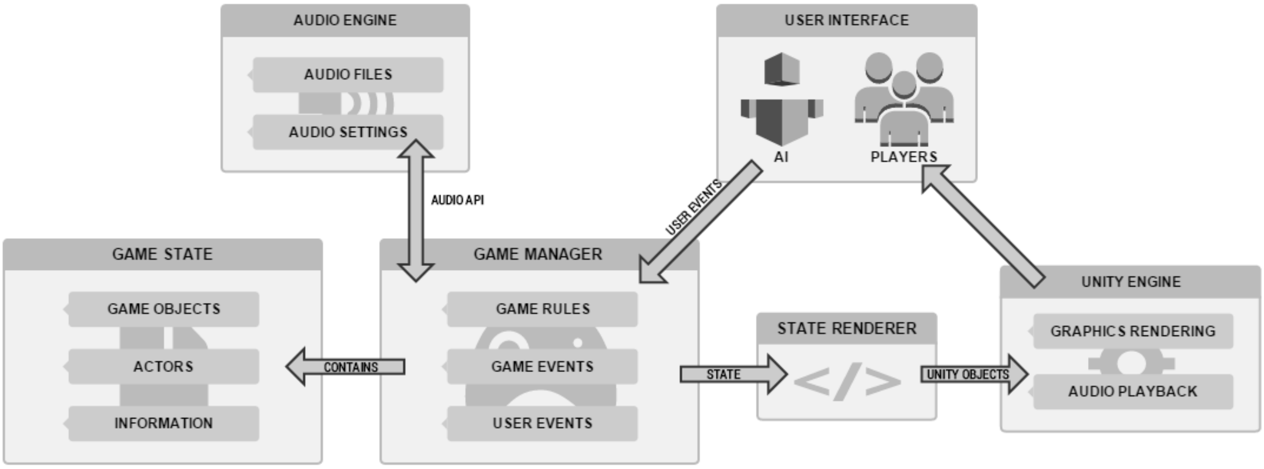

Figure 1: The architecture of the developed Tic-Tac-Toe game

instance of a game manager, allowing for rapid game development. Creating such stages becomes a nearly trivial matter as each successive stage of the game can be implemented by making slight alterations to the initial stage's game manager. The game manager operates by receiving input in the form of game and user events. Events are generated and sent to the game manager as a result of user actions or actor interactions. The game manager is responsible of handling events by applying all appropriate rules. The game rules contained within the game manager in combination with the capabilities of actors provide all the game logic required for a fully functional game. Finally the game manager is also responsible for supplying the current state to the state renderer whenever needed.

State Renderer The State Renderer transforms the game state so that it can be rendered by a graphic rendering and audio playback engine. In our current implementation each game object contained in the game state is mapped to one of our custom predefined Unity game objects. Besides generating the Unity game objects required, the state renderer also disposes of any obsolete objects when necessary.

User Interface The User Interface abstraction handles player actions to the game engine utilizing the appropriate game events. In our current implementation the user interface is based upon the Unity input manager, which allows receiving and acting upon a variety of user inputs.

The developed game engine offers a clear layer of abstraction that makes games portable to most graphic rendering and audio playback engines with minimal code alterations. The only parts of the platform that strictly depend on the Unity engine, at the time of writing, are the state renderer and the user interface.

Having described the game engine and related main concepts of our implementation, we now delve into a significant part of the added value of this work, which is the audio engine and the related sonic design.

3.2 Audio engine and sonic design

The developed audio engine was based on the environment of Unity Game Engine [2] and served as an API to audio utilization. In addition, the audio engine was responsible for managing the audio settings of the Tic-Tac-Toe game. An setting included, among other things, sets of sounds that were generated and engineered. Two sets of sounds were employed. One with auditory displays and one with earcons. For each, there were three placement options, presented later in this section. This led to 6 audio settings, all capable to be accessed and used by the application, through the audio engine.

Focusing on the audio design, there are two distinct tasks: the selection of proper sonic displays and the binaural processing of sounds. The former serves as the main audio interface between the player and game while the latter is used in order to effectively convey spatial information regarding the board employed in the Tic-Tac-Toe. Due the targeted audience, i.e. children, the selection and development of sonic displays was based on the appeal that the sounds would have. Due to this, the semantic context of the sonic displays was sought in special effects emerging from cartoons. More specifically, earcons were based on previously synthesized sounds utilized also in an audio only game of Tic-Tac-Toe [8]. Regarding auditory displays, 5 sounds were created from corresponding cartoon-like sounds. These are: i) "X" placement, ii) "O" placement, iii) boundary hit, iv) empty square, iv) wrong action, and v) movement to square. For displays i) and ii) the original sound was a spring-like audio effect. For the first one this was drawn from a knife hitting a wood and for the second a spring bouncing. For the boundary hitting, the sonic display was based on a ball hitting a

[2] http://unity3d.com

wooden object. Binaural processing of selected audio material was performed by the utilization of KEMAR HRTF library [10]. All used sounds were rendered for the following spatial positions:

1. 0° elevation and azimuths of 0, 45, 90, 135, 180, 225, 270, 315 degrees, where boundary hitting was rendered with azimuths 0, 90, 180 and 270 degrees

2. Elevation of 40°, 20° and -30° for the top, middle and bottom rows of board, respectively, and azimuths of 270, 0 and 45 degrees for the left, middle and right columns, respectively. Boundary hitting was rendered with elevation/azimuth pairs of 40/0, 20/60 and $-30/0$ and 20/300 degrees for the top, right, bottom and left boundary hitting, respectively.

3. Same as in 2 but with pitch shifting. Sounds for top and bottom raw were pitch shifted compared to sounds in middle raw. For the top one, all sounds were shifted one octave up where for the bottom raw all sounds were shifted one octave down. Boundary hits were same as in 2

Figure 2 presents the first case of spatial positioning whereas Figure 3 the remaining one.

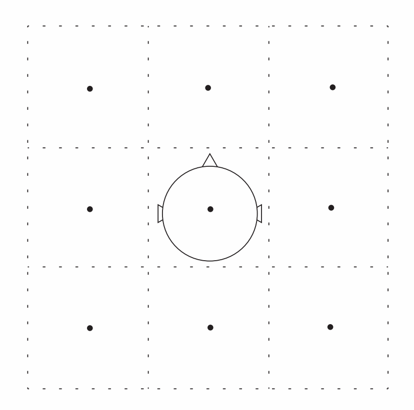

Figure 2: Illustration of the first case of sound spatial positioning

3.3 Tic-Tac-Toe Implementation

To accommodate for the accessibility needs of as many people as possible and tackle the challenge of partially impaired players, we used a simplistic, accessible graphical user interface (GUI). This GUI uses big visual elements and strong contrast. The resulting game is not visually impressive but allows users with severe eyesight restrictions to play it visually, if they can and want to support the auditory communication.

The audio interface focuses on providing users with direct feedback of their actions and their effects. There is no background music in order to give more emphasis to the individual sounds that describe the game state. Most importantly the audio engine implemented provides localized versions of

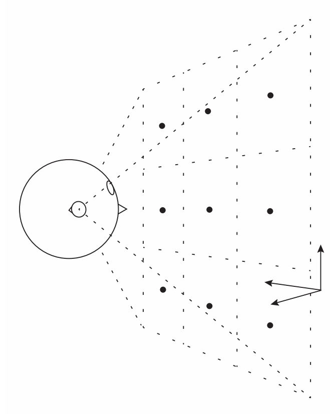

Figure 3: Illustration of the second and third cases of sound spatial positioning

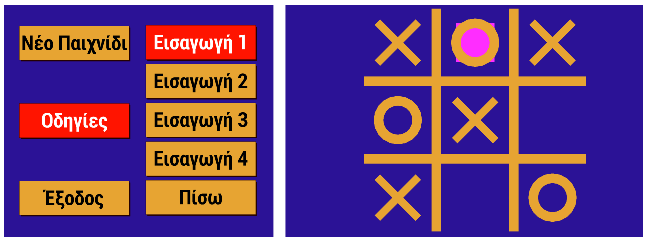

Figure 4: The visual user interface of the Tic-Tac-Toe game

each sound in order to give the players a sense of three dimensional space and direction. The localization is based on binaural audio algorithms that create realistic representations of each sound and its position relative to the player. In Figure 4 is a screenshot of the Tic-Tac-Toe's visual user interface.

In order to ensure that the game is playable by any player, disregarding previous experience or guidance, each game component contains recorded audio instructions. To face the challanges of game space, game state and rules understanding, four different tutorial games of increasing difficulty were implemented. Each tutorial game utilizes pre-recorded audio messages for guidance and aims to either teach the player part of the game of Tic-Tac-Toe or to help the player interpret the rest of the audio feedback.

In the following paragraphs we elaborate on the lifecycle of our use cases and the related findings.

4. USE CASE

The presented Tic-Tac-Toe game was evaluated in a real-world scenario. A set of interviews and user tests were performed on-site with visually impaired children. The tests were performed on specialized schools for the visually impaired. The tests were conducted with the help of a small group of children, which was sufficient for qualitative evaluation of the game, as described below. These interviews were conducted in three points in time: before starting development, after reaching our milestone, and after finishing our implementation.

One major, early finding in the user study was that blind children experience a severe lack of appropriately designed games. Our study gave the chance to many of the participating children to experience an accessible game for the first time. In particular, more than 12 visually impaired children participated from which 8 were legally considered blind. In the following sections we discuss the preliminary results that came before and during development, as well as the final results after presenting our finished Tic-Tac-Toe implementation.

4.1 Preliminary Results

The initial interviews explored the need for computer games for visually impaired children, as well as those children's preferences and computer-related knowledge. As mentioned above, the need for accessible computer games became immediately apparent. Most of the participating children had no previous experience using a computer for entertainment, besides listening to music. Even though our research shows the existence of some accessible games, it appears that during the last few years not enough effort has been put into building new games or dissemnitating existing ones to the corresponding communities. This meant that our users could be clearly considered as minimally knowledgeable on both computer use and computer games.

The two platforms that some of the children seemed to be somewhat familiar with were tablets and computers. At this point it was very difficult to explore actual game preferences, as the participating children found it a challenge to even imagine how a computer game of Tic-Tac-Toe could be made accessible to them. During these interviews we used a wooden Tic-Tac-Toe board to help describe the game to children, since most of them were completely unfamiliar with the specific game. After our initial interviews, we decided to target our implementation to desktop computers. The main advantage of desktop computers is that most children have access to one either at home or at school, while another important advantage is the big screen size and resolution that helps accommodate for accessible visual interfaces for the visually impaired and partially blind.

During our interviews after our first and second milestones we gathered feedback from a a subgroup of the user children, showing them the current stage of the game and exploring their thoughts on it. The most important result from these interviews was that, despite the fact that they had been introduced to the game during the first interviews, children needed a lot of help to understand how they to navigate in the 3 by 3 grid world of Tic-Tac-Toe. Our tutorials, at that point, were mostly aimed at teaching the rules and, thus, had not faced the navigation challenge. Guidance was also needed on how to interpret the audio clues of the game. Finally, there was also a very clear need for direct sound feedback from all their actions, in order to avoid becoming lost in the game world.

4.2 Final Results

After redesigning parts of the game and creating four introductory levels (or mini-games), we completed our implementation and collected the final evaluation. For our final user studies we followed a stricter procedure. Due to the use of headphones, there were no special needs or requirements for the audio environment. The evaluation process was, thus, conducted in an ordinary office using two of the schools general-purpose desktop computers. In both visits the following procedure was kept:

1. Each child was asked some initial questions regarding their background, knowledge of the game Tic-Tac-Toe and affinity with computers.
2. Each child was given headphones and instructed to follow the audio tutorials to learn how to play. In this step we only provided external help when required, in order to maintain a uniform time frame of about 25 minutes for all tutorials.
3. A mini-game session was conducted without any external help against a relatively difficult computer opponent. Thus, each child had the opportunity to play the developed Tic-Tac-Toe as it would normally do, if they owned this game.
4. Finally, when the game session ended, each child rated its experience and also participated in a short interview (5 minutes typical length).

Due to time limitations we occasionally had to supply some help but overall the children were able to finish the tutorials and move on to the next game on their own quite easily. During the interviews we collected several interesting comments as well as ratings on how interesting and usable the game was. Overall the children responded very positively:

- They rated the game with an average mark of 9.3 out of 10 to the question “Was the game interesting to you?”.
- They rated the game with an average mark of and 8.1 out of 10 to the question “Did you find the interface of the game usable?”.

The comments we gathered regarding improvements were as follows:

- We used the keyboard return key instead of the numpad return key, making it difficult for them to locate the keys during their turn.
- It was relatively difficulty at times to pinpoint the exact sound position during the opponent's moves and, thus, more interaction was required to update the cognitive representation of the game board. This indicated that further engineering was required with respect to the binaural rendering, based also on comments related to the duration of the sound (e.g. long sounds may be easier to communicate positioning) and other properties.

5. DISCUSSION AND CONCLUSIONS

There are several limitations that must be kept in mind when developing a game for blind children. Most importantly there needs to be constant audio feedback of anything that happens. The sounds used must be clearly distinguishable and able to communicate concurrent information. To allow the users to actually navigate the game space, sounds need to convey localization information. When using binaural audio it is also important to note that, as we realized during our final interviews, it doesn't work equally well for all sounds and positions.

Perhaps the most important limitation, however, is the difference in the spatial awareness of visually-impaired people. During the interviews we intentionally avoided describing the game space in order to encourage the children to create their own spatialization. As blind children develop their spatial awareness through their unique experiences, it can be significantly different to that of seeing people or other blind children. This can significantly increases the game difficulty in many cases. This problem can and should be addressed through specialized tutorials or other forms of specialized guidance, as we discovered through our use cases. Finally, it is important to note that children that have not been trained to use computers, using text-to-speech interfaces, do not have an understanding of navigation based on menus and sub-menus, thus such structures must be either avoided or explained as part of the game introduction.

Many blind children appear to have very limited access to computer games, even nowadays. Although many of them were completely negative at the idea that a computer game can work only with sounds at the beginning, they surprised us with great reviews and excitement once they tried the completed version. Although this is only a first step we believe that our implementation and research on this field can become a great starting point for further work. To further strengthen this field, we will make the development components available under open source, liberal licenses that allow reuse even in commercial settings.

We strongly believe that building games for children with and without disabilities can function in several ways:

- An accessible game can provide entertainment to children with disabilities, limiting the gap between children of different disability levels.
- Serious games can be built using accessibility-aware game engines and designs, facilitating tutors and children in learning processes. We plan to create a second game with a serious (i.e. learning) orientation in the next few months.
- Accessible games can be targetted to non-disabled children (and even adults), allowing them to increase their empathy and understanding of their disabled friends.

Acknowledgements

The work described in this paper was funded by the Greek NGO Programme “We are all Citizens” which is part of the EEA & Norway Grants for Greece. The Bodossaki Foundation is the Fund Operator of this Programme. The Programme aims to strengthen civil society and enhance the contribution of NGOs to social justice, democracy and sustainable development.

6. REFERENCES

[1] M. M. Blattner, D. A. Sumikawa, and R. M. Greenberg. Earcons and icons: Their structure and common design principles. *Hum.-Comput. Interact.*, 4(1):11–44, Mar. 1989.

[2] J. Desloge, W. Rabinowitz, and P. Zurek. Microphone-array hearing aids with binaural output .i. fixed-processing systems. *Speech and Audio Processing, IEEE Transactions on*, 5(6):529–542, 1997.

[3] K. Drossos, A. Floros, A. Giannakoulopoulos, and N. Kanellopoulos. Investigating the impact of sound angular position on the listener affective state. *Affective Computing, IEEE Transactions on*, PP(99):1–1, 2015.

[4] K. Drossos, A. Floros, and K. Koukoudis. Gestural user interface for audio multitrack real-time stereo mixing. In *Proceedings of the 8th Audio Mostly Conference*, AM '13, pages 2:1–2:6, New York, NY, USA, 2013. ACM.

[5] K. Drossos, S. Mimilakis, A. Floros, and N. Kanellopoulos. Stereo goes mobile: Spatial enhancement for short-distance loudspeaker setups. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 432–435, July 2012.

[6] Epicycle. Blindside - the audio adventure game. Electronically - Video Game, Feb 2015.

[7] Y. Eriksson and D. Gärdenfors. Computer games for children with visual impairments. *International Journal on Disability and Human Development*, 4(3):161–168, May 2011.

[8] A. Floros, N.-A. Tatlas, and S. Potirakis. Sonic perceptual crossings: A tic-tac-toe audio game. In *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound*, AM '11, pages 88–94, New York, NY, USA, 2011. ACM.

[9] J. Friberg and D. Gärdenfors. Audio games: New perspectives on game audio. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '04, pages 148–154, New York, NY, USA, 2004. ACM.

[10] B. Gardner and K. Martin. Hrtf measurements of a kemar dummy-head microphone. Technical Report 280, MIT Media Lab Perceptual Computing, 1994.

[11] W. W. Gaver. Auditory icons: Using sound in computer interfaces. *Hum.-Comput. Interact.*, 2(2):167–177, June 1986.

[12] M. Jeub and P. Vary. Binaural dereverberation based on a dual-channel wiener filter with optimized noise field coherence. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4710–4713, 2010.

[13] G. Kramer, B. Walker, T. Bonebright, P. Cook, J. Flowers, N. Miner, and J. Neuhoff. Sonification report: Status of the field and research agenda. White paper Master, National Science Foundation, 1998.

[14] N. Moustakas, A. Floros, and N. Grigoriou. Interactive audio realities: An augmented / mixed reality audio game prototype. In *Audio Engineering Society Convention 130*, May 2011.

[15] L. V. Nickerson and T. Hermann. Interactive sonifcation of grid-based games. In *Proceedings of the*

Audio Mostly Conference - A Conference on Interaction with Sound, pages 27–34, 2008.

[16] J. Sánchez, A. Tadres, A. Pascual-Leone, and L. Merabet. Blind children navigation through gaming and associated brain plasticity. In *Virtual Rehabilitation International Conference, 2009*, pages 29–36, June 2009.

[17] Somethinelse. Papa sangre. Electronically - Video Game, Feb 2015.

[18] F. Winberg and S. O. Hellström. Investigating auditory direct manipulation: Sonifying the towers of hanoi. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '00, pages 281–282, New York, NY, USA, 2000. ACM.

[19] B. Yuan and E. Folmer. Blind hero: Enabling guitar hero for the visually impaired. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 169–176, New York, NY, USA, 2008. ACM.